

## **REMARKS**

Claims 1-12, 14-32 and 34-92 remain pending in the application. Reconsideration is respectfully requested in light of the following remarks.

### **Double Patenting Rejections:**

The Office Action provisionally rejected claims 1, 3, 8-10, 14, 17, 45, 47, 52-54, 56, 57, 69, 71, 76-78, 80 and 81 under the judiciary created doctrine of obviousness-type double patenting as allegedly being unpatentable over claims 20, 23, 26-28, 35, 37, 38, 41, 44-46, 53 and 55 of U.S. Patent 7,831,693. The Office Action rejected claims 18-28, 30 and 58-68 under the judiciary created doctrine of obviousness-type double patenting as allegedly being unpatentable over claims 3, 6, 7, 9, 15, 16, 18-23, 25, 32-34, 36, 41, 42, 44-49, 51, 54-56, 58, 63, 64, 66-71 and 73 of U.S. Patent No. 7,698,398. Once the application is in condition for allowance, Applicant will consider filing a terminal disclaimer to obviate the obviousness-type double patenting rejections.

### **Section 103(a) Rejections:**

The Office Action on pages 7-8 rejected claims 1, 2, 5, 6, 8-10, 14-17, 45, 46, 49, 50, 52-54, 56, 57, 69, 70, 73, 74, 76-78 and 80 under 35 U.S.C. § 103(a) as allegedly being unpatentable over Curry et al. (U.S. Publication 2003/0233631) (hereinafter “Curry”) in view of Epionet, “Epiowave” (hereinafter “Epionet”) further in view of Huang et al. (“A Web Services-Based Framework for Business Integration Solutions”) (hereinafter “Huang”) and further in view of Olsen (U.S. Publication 2004/0243583) (hereinafter “Olsen”). Applicant respectfully traverses this rejection for at least the following reasons.

**In regard to claim 1, as a first matter, the cited references, alone or in combination, do not teach Applicant’s claim when viewed as a whole.**

The cited references, alone or in combination, do not teach program instructions that are executable to implement a Web Services architecture design service for generating integrated Web Service architectures for integrating Web Services with business systems, wherein the program instructions are executable to perform all of the elements combined and related as recited in claim 1.

Curry teaches a “Web services development method” (Title). Curry’s Abstract describes “A method for rapid design, development, deployment and support of web applications based on web services with minimum customized programming, maximized reuse of software components and compliance with standard development frameworks.” Curry’s FIG. 1 illustrates “a flow chart illustrating the steps of a rapid web services development method according to an illustrative embodiment of the present invention.” (paragraph [0030]). However, nowhere does Curry teach that the method illustrated in FIG. 1 is a computer-implemented method, and Curry does not describe a computer system that implements the method illustrated in FIG. 1. At most, Curry describes that particular steps or sub-steps of Curry’s method may be assisted by or performed using existing software tools. For example, in paragraph [0072], cited by the Office, Curry states: “In an illustrative embodiment of the invention, the Epiowave™ environment available from the applicant, Epionet Corporation of Dublin, Ireland, serves as the D&D environment in the D&D application creation step 32.” As another example, in paragraph [0076], also cited by the Office, Curry states (emphasis added): “In an illustrative embodiment of the invention, a toolset called MindManager by MindJet LLC of Sausalito, Calif. is used to create mind-maps which assist in the steps of creating the Role Control Diagram 44 and performing the Use Case Analysis 46.”

On page 8 of the Office Action of June 24, 2010, the Office admits that Curry does not teach implementing a Web Services architecture design service to generate integrated Web Service architectures for integrating Web Services with business systems, and asserts that it would have been obvious that “the system of Curry would include a Web Services architecture design service since the Epiowave toolset used in Curry is capable of planning, prototyping, testing, developing, and deploying web services.” The

Office further argues this assertion in the Response to Arguments, 72, on pages 36-37. **However, as noted above, Curry only describes that the “the D&D application creation step 32” of FIG. 1 may be performed using Epiowave.** Curry does not describe that other aspects of Curry’s system relied upon by the Office are performed using Epiowave or any other automated system. Moreover, the Office has not explained how a combination of Epiowave with Curry would result in anything but the system that Curry already teaches, for example as illustrated in FIG. 1. Neither Curry nor any combination of the cited art teaches a computer-implemented system that implements all of the elements of the method as illustrated in Curry’s FIG. 1, nor does Curry or any combination of the cited art teach a Web Services architecture design service that performs all of the elements as recited in claim 1 of the instant application.

The Epionet/Epiowave references, cited by the Office to support the assertion that Curry teaches the invention as claimed, may broadly assert that the toolset may provide for “planning, prototyping, testing, developing, and deploying web services;” however, the references do not teach that the Epionet/Epiowave platform, alone or in combination with the other references, performs the elements of the subject matter as recited in claim 1 for generating integrated Web Service architectures for integrating Web Services with business systems.

In (76), pages 39-41, of the Response to Arguments section of the Office Action of January 20, 2011, the Office disagrees with Applicant’s above arguments regarding the Curry and Epiowave references. The Office asserts, on page 40, that “the Epiowave software as described in Curry and Epionet/Epiowave provide the programmable instructions to perform the steps because Examiner interprets the programmable instructions to generate the use cases, high-level architectures, logical structures, etc., as programmable instructions that generate the use cases, high-level architectures, logical structures, etc. based on user input, so a user may design the use case, high-level architecture according to the method of Curry, and the programmable instructions provided by Epiowave are used by the user to generate a digital representation of the artifacts according to the method of Curry. As such, Examiner believes that Curry meets

these limitations because even though user input is required for designing the use cases, context diagrams, etc., the ultimate product is a software artifact in digital format (see [0172]) and modeling software tools, code development software tools, and deployment tools are used to model, develop and deploy the application.”

Applicant strongly disagrees with the Office’s interpretations. Curry, paragraphs [0165]-[0173], is describing “Development & Deployment Application Creation,” element 32 in Curry’s Fig. 1, which is “described in more detail with respect to Fig. 17.” Curry, in paragraphs [0071]-[0072], teaches:

“If it is determined that the application design is sufficiently optimized for code reuse and re-usability, the designed application is completed in a development and deployment (hereinafter “D&D”) environment application creation step 32. At this stage of web service development, the developers use a D&D environment to rapidly form the various functional classes, components, and presentation templates into a deployable application.

In an illustrative embodiment of the invention, the Epiowave™ environment available from the applicant, Epionet Corporation of Dublin, Ireland, serves as the D&D environment in the D&D application creation step 32. After the D&D application creation step 32 is performed, the application is typically ready for deployment 36. In the illustrative embodiment, the D&D environment also acts as a deployment utility to enable rapid deployment of the completed web service.”

Thus, Curry only relies on Epiowave to “complete” the previously designed application and to deploy the completed web service. “Completing” the application involves creating the application from the previously created design. **However, Curry does not teach that the various steps in Curry’s process for creating the design that are relied upon by the Office are performed using Epiowave or any other automated system.** Curry does not teach or suggest, alone or in combination with the other references, program instructions executable by a processor to generate one or more Use Cases for an integrated Web Service, generate a high-level architecture for the integrated Web Service in accordance with the one or more Web Services integration design patterns, and generate a logical architecture for the integrated Web Service.

Applicant further notes that the Office asserts that “Applicant appears to argue that the claim as recited requires a completely automated computer process to generate a Web Service architecture that does not require any user input.” Applicant notes that claim 1 neither requires nor rules out user input in the recited limitations. However, whether claim 1 requires or does not require user input is not relevant to Applicant’s arguments regarding the cited references. Curry, alone or in combination with the other references, does not teach or suggest program instructions executable by a processor to generate one or more Use Cases for an integrated Web Service, generate a high-level architecture for the integrated Web Service in accordance with the one or more Web Services integration design patterns, and generate a logical architecture for the integrated Web Service, as recited in claim 1, whether or not user input is required in these limitations.

Huang teaches a “Web services-based framework for business integration solutions” (Title). In Section 3, at page 18, second column, first full paragraph, Huang teaches “In this section, we propose a framework to develop Web services-based business integration solutions.” However, Huang only conceptually describes this framework, and does not teach or even suggest a computer implementation of the framework for developing Web services-based business integration solutions.

Olsen, on the other hand, describes systems and methods for providing web services in which an operator is prompted for information corresponding to a service, and a web service is automatically constructed in response to receiving the information (see, e.g., paragraphs [0004] and [0025]. However, Olsen’s systems operates by maintaining generic web service(s) and generating customized web services according to the provided information. (See, e.g., paragraphs [0023]-[0024]; e.g., paragraph [0023] states in part “Once the generic web service system receives information corresponding to a desired domain, the generic web service system can prompt the service provider for additional information until a web service, e.g., web service 114, can be constructed for the service provider.”) While Olsen does describe the automatic construction of a web service (e.g., “a web service is automatically constructed in response to receiving the information”);

however, Olsen only teaches that generic web services are customized according to input information and does not teach program instructions executable by a processor to generate integrated Web Service architectures for integrating Web Services with business systems, as recited in claim 1.

In contrast to the cited references, claim 1 recites a system for integrating Web Services with business systems, comprising: a processor; and a memory comprising program instructions, wherein the program instructions are executable by the processor to implement a Web Services architecture design service configured to generate integrated Web Service architectures for integrating Web Services with business systems, wherein, to generate an integrated Web Service architecture for integrating a specific Web Service with a specific business system, the program instructions are executable by the processor to: generate the integrated Web Service architecture comprising a plurality of heterogeneous components of the specific business system in accordance with one or more Web Services integration design patterns for integrating Web Services with business systems; wherein, to generate the integrated Web Service architecture, the program instructions are executable by the processor to...<snip the elements performed by the program instructions>...and provide output indicating the generated integrated Web Service architecture for integrating the specific Web Service with the specific business system.

Furthermore, at paragraph [0022], Curry teaches (emphasis added): “Interface templates are also reviewed for compliance with a standard framework architecture.” In paragraph [0055], Curry teaches (emphasis added): “The framework as referred to in the present invention is an overall architecture which provides a template for building enterprise web solutions. The framework includes a pre-built architecture that allows developers to rapidly create applications based on business components and web services.” Thus, Curry does not teach a system comprising program instructions executable to implement a Web Services architecture design service configured to generate an integrated Web Service architecture for integrating a specific Web Service with a specific business system. Instead, Curry clearly teaches a “standard framework

architecture” that clearly pre-exists and that may allegedly be used to “rapidly create applications based on business components and web services.”

On page 37 of the Response to Arguments section of the Office Action of June 24, 2010, the Office argues that “a specific instance of the framework template must be generated for the web service under development.” However, this “specific instance” would still be just a copy of Curry’s “standard framework architecture” that may be modified, and does not teach a system comprising program instructions executable to implement a Web Services architecture design service configured to generate an integrated Web Service architecture for integrating a specific Web Service with a specific business system, as recited in Applicant’s claims.

In (77), page 41, of the Response to Arguments section of the Office Action of January 20, 2011, the Office disagrees with Applicant’s above arguments regarding the Curry and Epiowave references. The Office asserts that Curry’s “pre-built framework is used as a blueprint or generic application based upon which the specific application is generated and additional customization is added to the framework template for the specific web service under development (see [0055]). Since the framework template is a software artifact, any customization performed for the framework template would be performed by computer instructions to reflect the changes in the software artifact.” However, the Office’s assertion that “Since the framework template is a software artifact, any customization performed for the framework template would be performed by computer instructions to reflect the changes in the software artifact” is clearly conclusory, and not supported by any evidence of record. The Office is asserting something in hindsight that is not disclosed anywhere in the references.

**Moreover, Applicant’s claim 1 is not directed to performing customization of a framework template, but is instead directed to a system comprising program instructions executable to implement a Web Services architecture design service configured to generate an integrated Web Service architecture for integrating a specific Web Service with a specific business system.**

Curry, alone or in combination with Epiowave and the other references, does not teach a system comprising program instructions executable to implement a Web Services architecture design service configured to generate an integrated Web Service architecture for integrating a specific Web Service with a specific business system, as recited in Applicant's claims.

**As can be seen from above, the cited references, alone or in combination, clearly do not teach a system as recited in Applicant's claim 1 when the claim is viewed as a whole.**

The Office is clearly attempting a piecemeal reconstruction of Applicant's invention in hindsight without considering the claimed invention as a whole. Such reconstruction is improper. *See, e.g., Interconnect Planning Corp. v. Feil*, 774 F.2d 1132, 1143, 227 USPQ 543, 551 (Fed. Cir. 1985) (it is insufficient to select from the prior art the separate components of the inventor's combination, using the blueprint supplied by the inventor); *Uniroyal, Inc. v. Rudkin-Wiley Corp.*, 837 F.2d 1044, 1051-52, 5 USPQ 2d 1434, 1438 (Fed. Cir. 1988) (it is impermissible to reconstruct the claimed invention from selected pieces of prior art absent some suggestion, teaching, or motivation in the prior art to do so). The Office cannot use the claimed invention as an instruction manual or "template" to piece together the teachings of the prior art so that the claimed invention is rendered obvious. *In re Fritch*, 23 USPQ 2d 1780, 1784 (Fed. Cir. 1992). "One cannot use hindsight reconstruction to pick and choose among isolated disclosures in the prior art to deprecate the claimed invention." *In re Fine*, 837 F.2d 1071, 1075, 5 USPQ 2d 1596, 1600 (Fed. Cir. 1988).

**In further regard to claim 1, the cited references, alone or in combination, do not teach program instructions that are executable to implement a Web Services architecture design service for generating integrated Web Service architectures for integrating Web Services with business systems.**



On page 8 of the Office Action of January 20, 2011, the Office asserts “ a processor (see [0072], [0076], 0176]); software toolsets such as Epiowave is used to integrate the web service and it is well known in the art that software is executed by a processor); and a memory comprising program instructions (see [0072], [0076], 0176]); software toolsets such as Epiowave is used to integrate the web service and it is well known in the art that software is stored in memory before being executed by a processor) to generate integrated Web Service architectures....” However, Applicant notes that claim 1 does not merely recite “a memory comprising program instructions to generate integrated Web Service architectures...” as implied by the Office. Claim 1 specifically recites “a memory comprising program instructions, wherein the program instructions are executable by the processor to implement a Web Services architecture design service configured to generate integrated Web Service architectures for integrating Web Services with business systems.” Applicant assert that program instructions executable by a processor to implement a Web Services architecture design service configured to generate integrated Web Service architectures for integrating Web Services with business systems as recited in Applicant’s claim 1 is not well known in the art. Applicant notes that the claimed limitations are directed to the invention as a whole and assert that the above-mentioned limitations in light of the claimed invention as a whole is not well known (see Applicant’s other arguments).

Furthermore, the Office’s assertion “software toolsets such as Epiowave is used to integrate the web service” is not commensurate with the actual teachings of the Curry reference. Curry does not describe that the aspects of Curry’s system relied upon by the Office to allegedly teach the limitations as recited in Applicant’s claim 1 are performed using Epiowave or any other automated system. Curry, in paragraphs [0071]-[0072], teaches:

“If it is determined that the application design is sufficiently optimized for code reuse and re-usability, the designed application is completed in a development and deployment (hereinafter “D&D”) environment application creation step 32. At this stage of web service development, the developers use a D&D environment to rapidly form the various functional classes, components, and presentation templates into a deployable application.

In an illustrative embodiment of the invention, the Epiowave™ environment available from the applicant, Epionet Corporation of Dublin, Ireland, serves as the D&D environment in the D&D application creation step 32. After the D&D application creation step 32 is performed, the application is typically ready for deployment 36. In the illustrative embodiment, the D&D environment also acts as a deployment utility to enable rapid deployment of the completed web service.”

Thus, contrary to the Office’s assertion, Curry only relies on Epiowave to “complete” the previously designed application and to deploy the completed web service (step 32 in Curry’s process as illustrated in Fig. 1). “Completing” the application involves creating the application from the previously created design. **However, Curry does not teach that the various steps in Curry’s process for creating the design that are relied upon by the Office are performed using Epiowave or any other automated system.** Since Curry does not teach that the various steps in Curry’s process relied upon by the Office for creating the design are performed using Epiowave or any other automated system, Curry cannot be said to teach program instructions executable by a processor to implement a Web Services architecture design service configured to generate integrated Web Service architectures for integrating Web Services with business systems as recited in claim 1.

**In further regard to claim 1, the cited references do not teach at least the features of program instructions executable by a processor to generate one or more Use Cases for the integrated Web Service in accordance with the one or more Web Services integration design patterns, wherein each Use Case models a particular business scenario for the integrated Web Service, as recited in claim 1.**

The Office relies on Curry to teach “generate one or more Use Cases for the integrated Web Service,” citing Curry, paragraphs [0079]-[0084]. These paragraphs describe a step 46 in Curry’s FIG. 2, which expands on element 10 (“Business Logic Development”) of Curry’s FIG. 1 (see paragraph [0074]). At paragraph [0074], Curry states “During the business logic development step 10, a precise description of the customer’s business requirements or the business logic requirements of the web services

under development are compiled by performing an orderly sequence of steps.” In paragraph [0076], Curry states “First, a set of Role Control Diagrams 44 is developed and a Use Case Analysis is performed 46.” However, Curry does not teach that this Use Case Analysis generates one or more Use Cases for an integrated Web Service in accordance with the one or more Web Services integration design patterns, as recited in claim 1.

The Office relies on Huang to teach integration design patterns, citing page 18, second column, paragraph 2 and page 20, first column paragraphs 2 and 3 and second column, paragraphs 1-3. In Section 3, at page 18, second column, first full paragraph, Huang teaches “In this section, we propose a framework to develop Web services-based business integration solutions.” In the first paragraph cited from page 20, Huang teaches “In essence, this framework uses four different design patterns...” However, Huang does not teach generating one or more Use Cases for an integrated Web Service in accordance with the one or more Web Services integration design patterns, as recited in claim 1.

Moreover, the other references fail to overcome this shortcoming, and no combination of the references teaches this subject matter as recited in claim 1.

In (78), pages 41-42, of the Response to Arguments section of the Office Action of January 20, 2011, the Office asserts “it is well known in the art that design patterns provide a template for solving a problem and software engineers widely use design patterns at different stages of software development life cycle to solve problems following that template. There is no restriction on when the design pattern can be employed in a development cycle...Therefore, Examiner believes that it is obvious that one of ordinary skill in the art...developing use cases for a particular problem (i.e., web service with business integration) would have the necessary skills to use known design patterns during the development stage for that particular problem.”

Applicant asserts that these claim limitations as recited in claim 1 are not well known the art. Applicant notes that claim 1 does not simply recite using known design patterns during a development stage, but instead recites generating one or more Use

Cases for an integrated Web Service in accordance with one or more Web Services integration design patterns, wherein each Use Case models a particular business scenario for the integrated Web Service. Applicant asserts that generating one or more Use Cases for an integrated Web Service in accordance with one or more Web Services integration design patterns, wherein each Use Case models a particular business scenario for an integrated Web Service, is not well known. Moreover, **the Office's assertion is not supported by any evidence of record.** Merely stating that individual aspects of a claimed invention are well known does not render the combination well known without some objective reason to combine the individual teachings. *Ex parte Levengood*, 28 USPQ2d 1300. MPEP 2144.03 states "Official notice without documentary evidence to support an examiner's conclusion is permissible only in some circumstances...Official notice unsupported by documentary evidence should only be taken by the examiner where the facts asserted to be well-known, or to be common knowledge in the art are capable of instant and unquestionable demonstration as being well-known." Applicants therefore respectfully request the Examiner to provide either references that teach the claimed limitations of Applicants' application or an affidavit attesting to the Examiner's personal knowledge of same.

**In further regard to claim 1, the cited references do not teach at least the features of program instructions executable by a processor to generate a high-level architecture for the integrated Web Service in accordance with the one or more Web Services integration design patterns, wherein the high-level architecture identifies two or more entities of the integrated Web Service and the relationships and interactions among the entities, as recited in claim 1.**

The Office relies on Curry, citing paragraph [0097] and asserting "the context diagram describes the high-level architecture." Paragraph [0097] states (emphasis added): "Once the various forms of project descriptions, including business rules 52, state diagrams 54, swim-lane diagrams 56 and activity diagrams 58 are complete, a context diagram 60 is prepared." However, Curry does not teach that this context diagram is

prepared in accordance with the one or more Web Services integration design patterns, as recited in claim 1.

The Office Action relies on Huang to teach integration design patterns, citing page 18, second column, paragraph 2 and page 20, first column paragraphs 2 and 3 and second column, paragraphs 1-3. In Section 3, at page 18, second column, first full paragraph, Huang teaches “In this section, we propose a framework to develop Web services-based business integration solutions.” In the first paragraph cited from page 20, Huang teaches “In essence, this framework uses four different design patterns...” However, Huang does not teach generating a high-level architecture for the integrated Web Service in accordance with the one or more Web Services integration design patterns, as recited in claim 1.

Moreover, the other references fail to overcome this shortcoming, and no combination of the references teaches this subject matter as recited in claim 1.

In (79), pages 42-43, of the Response to Arguments section of the Office Action of January 20, 2011, the Office asserts “it is well known in the art that design patterns provide a template for solving a problem and software engineers widely use design patterns at different stages of software development to solve problems following that template...Therefore, Examiner submits that it is obvious that one of ordinary skill in the art...developing high level architecture for a particular problem (i.e., web service with business integration) would have the necessary skills to use known design patterns during the development stage for that particular problem.”

Applicant traverses the Office’s assertion that these claim limitations as recited in claim 1 are well known in the art. Applicant notes that claim 1 does not simply recite “developing high level architecture for a particular problem,” but instead recites generating a high-level architecture for an integrated Web Service in accordance with the one or more Web Services integration design patterns, wherein the high-level architecture identifies two or more entities of the integrated Web Service and the relationships and

interactions among the entities. Applicant asserts that generating a high-level architecture for an integrated Web Service in accordance with the one or more Web Services integration design patterns, wherein the high-level architecture identifies two or more entities of the integrated Web Service and the relationships and interactions among the entities, is not well known. Moreover, **the Office's assertion is not supported by any evidence of record.** Merely stating that individual aspects of a claimed invention are well known does not render the combination well known without some objective reason to combine the individual teachings. *Ex parte Levengood*, 28 USPQ2d 1300. MPEP 2144.03 states "Official notice without documentary evidence to support an examiner's conclusion is permissible only in some circumstances...Official notice unsupported by documentary evidence should only be taken by the examiner where the facts asserted to be well-known, or to be common knowledge in the art are capable of instant and unquestionable demonstration as being well-known." Applicants therefore respectfully request the Examiner to provide either references that teach the claimed limitations of Applicants' application or an affidavit attesting to the Examiner's personal knowledge of same.

**In further regard to claim 1, the cited references do not teach at least the features of program instructions executable by a processor to generate a logical architecture for the integrated Web Service according to the business scenarios modeled by the one or more Use Cases and in accordance with the one or more Web Services integration design patterns, as recited in claim 1.**

The Office relies on Curry, citing paragraph [0055]-[0059] and asserting "the framework structure including a number of layers is the logical architecture." However, Curry does not teach that the "framework structure" is generated according to the business scenarios modeled by the one or more Use Cases and in accordance with the one or more Web Services integration design patterns, as recited in claim 1.

Moreover, the other references fail to overcome this shortcoming, and no combination of the references teaches this subject matter as recited in claim 1.

In (80), pages 43-44, of the Response to Arguments section of the Office Action of January 20, 2011, the Office asserts “the framework structure is generated according to the business scenario modeled by the one or more Use Cases since the Use Cases and framework structure are all used to address the same problem and the framework structure step occurs after the Use Case step,” citing Fig. 1, [0051], [0055], and [0074]. First, the Office’s assertion is speculative. Applicant reminds the Examiner that mere speculation is not sufficient to support a *prima facie* case of obviousness. See *In re Warner*, 379 F.2d 1011, 1017, 154 USPQ 173, 178 (CCPA 1967); *In re Sporck*, 301 F.2d 686, 690, 133 USPQ 360, 364 (CCPA 1962). Moreover, Office is clearly attempting a piecemeal reconstruction of Applicant’s invention in hindsight. Such hindsight reconstruction is improper.

Furthermore, step 18 of Curry’s Fig. 1 as referenced in paragraph [0055] does not teach generating the framework structure, as alleged by the Office. Instead, step 18 and paragraph [0055] describe a “framework review.” Paragraph [0055] states (emphasis added): “The framework as referred to in the present invention is an overall architecture which provides a template for building enterprise web solutions. The framework includes a pre-built architecture...” **Curry’s “framework” as described in paragraph [0055] clearly already exists prior to Curry’s “framework review 18,” and is not “generated.”**

Thus, Curry’s teachings of a “framework structure” and “framework review” in paragraph [0055] clearly do not teach or suggest generating a logical architecture for the integrated Web Service according to the business scenarios modeled by the one or more Use Cases and in accordance with the one or more Web Services integration design patterns, as recited in claim 1.

In addition, in (80), pages 43-44, of the Response to Arguments section of the Office Action of January 20, 2011, the Office asserts “Huang teaches Web Service integration design patterns...it would have been obvious...that the framework structure

would be generated in accordance with one or more Web Service integration design patterns...” However, as noted above, Curry does not teach that the framework structure is generated at element 18 of Fig. 1 and paragraph [0055], instead teaching a “framework review” process applied to an existing framework structure. Moreover, the Office’s assertion is again speculative. The Office is clearly attempting a piecemeal reconstruction of Applicant’s invention in hindsight. Such hindsight reconstruction is improper.

**In further regard to claim 1, the cited references do not teach wherein the logical architecture identifies two or more logical components of the integrated Web Service and the relationship among the two or more logical components according to a plurality of integration tiers, and wherein the logical architecture comprises two or more layers, as recited in claim 1.**

The Office relies on Curry, citing paragraph [0055]-[0059] and asserting “the framework structure including a number of layers is the logical architecture.” However, Curry only teaches that the “framework structure” may include layers, and does not teach a logical architecture that identifies two or more logical components of the integrated Web Service and the relationship among the two or more logical components according to a plurality of integration tiers, and that teaches the logical architecture comprises two or more layers. Claim 1 recites both a plurality of integration tiers and two or more layers. Curry only teaches that the framework may include a number of layers, for example in paragraph [0056]: “The structure includes a number of layers or sub-architectures. Each layer addresses a specific area within a standard web application.” Curry does not teach a plurality of integration tiers and two or more layers, as recited in claim 1.

In the rejection of claim 2, the Office asserts that Curry teaches “defining a number of integration tiers,” citing paragraph [0151]. However, this citation refers to “the logical layers of the application,” which appear to be the same “layers” referred to in paragraph [0056].



In (81), page 44, of the Response to Arguments section of the Office Action of January 20, 2011, the Office asserts “Curry teaches that tier and layer are used interchangeably,” citing paragraph [0016]. However, this actually supports Applicants’ above arguments that Curry only teaches that the framework may include a number of layers. In contrast to Curry’s teachings, claim 1 recites both a plurality of integration tiers and two or more layers – two different and distinct limitations. Applicant notes that the specification and drawings of the instant application clearly distinguish between integration tiers and layers as used in Applicant’s claim 1. See, e.g., Fig. 22 and Fig. 64; page 111, lines 1-22. Applicant reminds the Office that claim language should be read in light of the specification as it would be interpreted by one of ordinary skill in the art. The Court of Appeals for the Federal Circuit has cautioned against an unreasonably broad claim construction:

Although the PTO emphasizes that it was required to give all “claims their broadest reasonable construction” [ ... ], this court has instructed that any such construction be “*consistent with the specification*, ... and that claim language should be read in light of the specification as it would be interpreted by one of ordinary skill in the art.” *In re Bond*, 910 F.2d 831, 833 (Fed.Cir.1990)...claims should always be read in light of the specification and teachings in the underlying patent. See *Schriber-Schroth Co. v. Cleveland Trust Co.*, 311 U.S. 211, 217 (1940).

*In re Suitco Surface, Inc.*, No. 2009-1418, 2010 WL 1462294, at \*4 (Fed. Cir. 2010).

The Office goes on to assert “Therefore, Curry teaches a plurality of integration tiers and two or more layers by teaching that the sub-architecture contain a number layers because the tiers and layers are used interchangeably in the Curry reference.” Applicant strongly disagrees; the correct conclusion would be just the opposite of what the Office asserts – Curry does not teach a plurality of integration tiers and two or more layers since the terms tiers and layers are used interchangeably in the Curry reference.

The Office goes on to speculate that “tiers and layers sometimes do have different meanings in the art with tiers referring to a physical separation of components and layer referring to a logical separation. However, even with this different definition for tiers and

layers, Curry still meets the limitation because Curry teaches logical separation (i.e., layers) and physical separation (i.e., application may be each physically separated or may be combined into components that include multiple logical layers, see [0151]).”

Applicant again notes that the specification and drawings of the instant application clearly distinguish between integration tiers and layers as used in Applicant’s claim 1. See, e.g., Fig. 22 and Fig. 64; page 111, lines 1-22. Even if Curry teaches “logical separation (i.e., layers) and physical separation” as asserted by the Office, Applicant again reminds the Office that claim language should be read in light of the specification as it would be interpreted by one of ordinary skill in the art, and that the Court of Appeals for the Federal Circuit has cautioned against an unreasonably broad claim construction. *In re Suitco Surface, Inc.*, No. 2009-1418, 2010 WL 1462294, at \*4 (Fed. Cir. 2010). The Office should rely on Applicant’s specification, and not on unsupported speculations as to other definitions for “tiers” and “layers” in the art, in interpreting the language of the claims.

**In further regard to claim 1, the Office has failed to establish a proper reason to combine the references.**

In regard to the proposed combination of Curry and Huang, the Office asserts that “it would have been obvious...to have modified Curry such that the web service architecture is generated in accordance with one or more integration design patterns as taught by Huang because design patterns are well known in the art and commonly used by programmers since design patterns provide the benefit of capturing a standard solution to a common programming problem for reuse.” However, Huang teaches four specific design patterns – a Composite pattern, a Mediator pattern, a Command pattern, and a State pattern. It is unclear as to how Curry’s system would be modified with these specific design patterns, much less how the modification could be made without changing the principle of operation of Curry’s “Web services development method” as disclosed. Curry and Huang each propose distinctly different methods for achieving similar results. Modifying Curry with Huang’s specific “design patterns” would appear to necessitate

significant changes in Curry's system. "If the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious." *In re Ratti*, 270 F.2d 810, 123 USPQ 349 (CCPA 1959). Thus, one of ordinary skill would not have combined the teachings of Huang with the teachings of Curry in the manner proposed by the Office. Accordingly, the Office has failed to establish a *prima facie* case of obviousness.

In the Response to Arguments, 77, on pages 37-38 of the Office Action of June 24, 2010, the Office asserts that it is "not clear...why incorporating the use of design patterns in the development method taught by Curry would necessitate significant changes." However, as Applicant has noted, Huang teaches four specific design patterns – a Composite pattern, a Mediator pattern, a Command pattern, and a State pattern. These design patterns are specifically intended for use in Huang's "Web services-based framework for business integration solutions." (Title). Curry and Huang each propose distinctly different methods for achieving similar results. It is unclear as to how Curry's system would be modified with these specific design patterns, much less how the modification could be made without changing the principle of operation of Curry's "Web services development method" as disclosed. Modifying Curry with Huang's specific "design patterns" would thus appear to necessitate significant changes in Curry's system. Moreover, while the Office broadly asserts that it is obvious that "one of ordinary skill in the art...would have the necessary skills to use design patterns for that particular problem," the Office provides no support as to how Curry's system would be modified with these specific design patterns, much less how the modification could be made without changing the principle of operation of Curry's "Web services development method" as disclosed.

In (82), page 45, of the Response to Arguments section of the Office Action of January 20, 2011, the Office asserts "Curry is directed to a Web service development method and Huang is directed to design patterns usable in web services. Examiner submits that it is well known in the art that design patterns provide a template for solving

a problem and that software engineers knows how to apply design patterns during software development. Curry is modified by Huang in that the design patterns taught by Huang are used in developing the web service according to the method of Curry.”

First, Applicant assert that, even if “it is well known in the art that design patterns provide a template for solving a problem and that software engineers knows how to apply design patterns during software development,” the claim limitations as recited in claim 1 are not well known the art. Moreover, the Office is simply repeating a previous assertion, and adds nothing new in response to Applicant’s actual arguments as stated in the previous responses. Applicants maintain that Huang teaches four specific design patterns that are specifically intended for use in Huang’s “Web services-based framework for business integration solutions.” (Title). Curry and Huang each propose distinctly different methods for achieving similar results. It is unclear as to how Curry’s system would be modified with these specific design patterns, much less how the modification could be made without changing the principle of operation of Curry’s “Web services development method” as disclosed. Modifying Curry with Huang’s specific “design patterns” would thus appear to necessitate significant changes in Curry’s system. The Office has provided no support as to how Curry’s system would be modified with these specific design patterns, much less how the modification could be made without changing the principle of operation of Curry’s “Web services development method” as disclosed.

In regard to the Curry and Epiowave references, the Office asserts it would have been obvious that “the system of Curry would include a Web Services architecture design service since the Epiowave toolset used in Curry is capable of planning, prototyping, testing, developing, and deploying web services.” However, at paragraph [0072] Curry states that “In an illustrative embodiment of the invention, the Epiowave™ environment available from the applicant, Epionet Corporation of Dublin, Ireland, serves as the D&D environment in the D&D application creation step 32.” Curry does not describe that other aspects of Curry’s system relied upon by the Office are performed using Epiowave. Moreover, the Office has not explained how a combination of Epiowave with Curry

would result in anything but the system that Curry already teaches. Accordingly, the Office has failed to establish a *prima facie* case of obviousness.

In (83), pages 45-46, of the Response to Arguments section of the Office Action of January 20, 2011, the Office asserts “Epionet/Epiowave provide the toolset to perform the steps as taught by Curry because Curry teaches that toolsets could be used to perform the steps (see [0176]). The method described by Curry involves the planning, prototyping, testing, developing, and deploying web services and therefore, Epiowave would be capable of supplying the necessary toolset functionalities for the method of Curry to develop the necessary artifacts.” However, **Curry only describes that the “the D&D application creation step 32” of FIG. 1 may be performed using Epiowave.** Curry does not describe that other aspects of Curry’s system relied upon by the Office are performed using Epiowave or any other automated system. Moreover, the Office has not explained how a combination of Epiowave with Curry would result in anything but the system that Curry already teaches, for example as illustrated in FIG. 1. Neither Curry nor any combination of the cited art teaches a computer-implemented system that implements all of the elements of the method as illustrated in Curry’s FIG. 1, nor does Curry or any combination of the cited art teach a Web Services architecture design service that performs all of the elements as recited in claim 1 of the instant application.\

Paragraph [0176] of Curry only states that “various steps in the present invention may be performed without any particular toolset or with various other toolsets that are available from other sources without departing from the spirit and scope of the present invention.” This fails to overcome the fact that Curry, alone or in combination with the other references including the Epionet/Epiowave references, fails to teach or suggest the system as recited in claim 1 when viewed as a whole.

**In further regard to claim 1, the cited references do not teach provide output indicating the generated integrated Web Service architecture for integrating the specific Web Service with the specific business system, as recited in claim 1.**

On page 10 of the Office Action of January 20, 2011, the Office admits that Curry, Epionet, and Huang do not teach the above limitations, and asserts that it would have been obvious that there would have been an output indicated the generated integrated Web Service architecture because “the template is used by developers for customization of enterprise solutions (see [0055] of Curry) and as it is well known in the art that a user must be provided with an indication that a template for a web service exists before the template can be customized (see [0005] of Olsen).” Applicant first notes that [0005] of Olsen does not teach that “a user must be provided with an indication that a template for a web service exists before the template can be customized.” Applicant further notes that, even if “a user must be provided with an indication that a template for a web service exists before the template can be customized” is well known in the art as asserted by the Office, claim 1 does not recite providing “an indication that a template for a web service exists before the template can be customized.” Instead, claim 1 recites providing output indicating a generated integrated Web Service architecture for integrating the specific Web Service with the specific business system.

Applicant further asserts that these claim limitations as recited in claim 1 are not well known the art. Merely stating that individual aspects of a claimed invention are well known does not render the combination well known without some objective reason to combine the individual teachings. *Ex parte Levengood*, 28 USPQ2d 1300. MPEP 2144.03 states “Official notice without documentary evidence to support an examiner’s conclusion is permissible only in some circumstances...Official notice unsupported by documentary evidence should only be taken by the examiner where the facts asserted to be well-known, or to be common knowledge in the art are capable of instant and unquestionable demonstration as being well-known.” Applicants therefore respectfully request the Examiner to provide either references that teach the claimed limitations of Applicants’ application or an affidavit attesting to the Examiner’s personal knowledge of same.

Thus, for at least the reasons presented above, the rejection of claim 1 is not supported by the cited art and removal thereof is respectfully requested. Similar remarks as those above regarding claim 1 also apply to claims 45 and 69.

Applicant also asserts that the rejection of numerous ones of the dependent claims under the 35 U.S.C. § 103(a) rejection is further unsupported by the cited art. However, since the rejection has been shown to be unsupported for the independent claims, a further discussion of the dependent claims is not necessary at this time.

The Office Action on page 14 rejected claims 3, 47 and 71 as being unpatentable over Curry in view of Epionet and further in view of Huang, Olsen and Curtis, et al. (U.S. Publication 2003/0115377) (hereinafter "Curtis"). However, since the rejection has been shown to be unsupported for the independent claims from which these claims depend, a further discussion of this rejection is not necessary at this time.

The Office Action on page 15 rejected claims 4, 48 and 72 under 35 U.S.C. § 103(a) as allegedly being unpatentable over Curry in view of Epionet and further in view of Huang, Olsen and Connell, et al. (U.S. Publication 2003/0074401) (hereinafter "Connell"). However, since the rejection has been shown to be unsupported for the independent claims from which these claims depend, a further discussion of this rejection is not necessary at this time.

The Office Action on page 16 rejected claims 7, 11, 12, 51, 55, 75 and 79 under 35 U.S.C. § 103(a) as allegedly being unpatentable over Curry in view of Epionet and further in view of Huang, Olsen and Chappell, et al. ("Java Web Services") (hereinafter "Chappell"). However, since the rejection has been shown to be unsupported for the independent claims from which these claims depend, a further discussion of this rejection is not necessary at this time.

The Office Action on page 18 rejected claims 18-24, 28, 29, 58-64, 68, 82-88 and 92 under 35 U.S.C. § 103(a) as allegedly being unpatentable over Curry in view of

Epionet and further in view of Siegel (“Using OMG’s Model Driven Architecture (MDS) to Integrate Web Services”), Huang and Olsen. Applicant traverses this rejection for at least the reasons indicated below.

**In regard to claim 18, as a first matter, the cited references, alone or in combination, do not teach program instructions executable to implement a Web Services architecture design service for generating integrated Web Service architectures.**

As noted above in regards to claim 1, Curry does not teach that the method illustrated in FIG. 1 is a computer-implemented method, and Curry does not describe a computer system that implements the method illustrated in FIG. 1. At most, Curry describes that particular steps or sub-steps of Curry’s method may be assisted by or performed using existing software tools. Huang only conceptually describes a framework for developing Web services-based business integration solutions, and does not teach or even suggest a computer implementation of the framework for developing Web services-based business integration solutions. The Epionet/Epiowave references may broadly assert that the toolset may provide for “planning, prototyping, testing, developing, and deploying web services,” however, the references do not teach that the Epionet/Epiowave platform performs the elements of the subject matter as recited in claim 18. Olsen only teaches that generic web services are customized according to input information and does not teach program instructions executable by a processor to generate integrated Web Service architectures for integrating Web Services with business systems.

On page 8 of the Office Action of June 24, 2010, the Office admits that Curry does not teach implementing a Web Services architecture design service to generate integrated Web Service architectures for integrating Web Services with business systems, and asserts that it would have been obvious that “the system of Curry would include a Web Services architecture design service since the Epiowave toolset used in Curry is capable of planning, prototyping, testing, developing, and deploying web services.” The Office further argues this assertion in the Response to Arguments, 72, on pages 36-37.



**However, as noted above, Curry only describes that the “the D&D application creation step 32” of FIG. 1 may be performed using Epiowave. Curry does not describe that other aspects of Curry’s system relied upon by the Office are performed using Epiowave or any other automated system. Moreover, the Office has not explained how a combination of Epiowave with Curry would result in anything but the system that Curry already teaches, for example as illustrated in FIG. 1. Neither Curry nor any combination of the cited art teaches a computer-implemented system that implements all of the elements of the method as illustrated in Curry’s FIG. 1, nor does Curry or any combination of the cited art teach a Web Services architecture design service that performs all of the elements as recited in claim 18 of the instant application.**

In contrast to the cited references, claim 18 recites a system for generating integrated Web Service architectures, comprising: a processor; and a memory comprising program instructions, wherein the program instructions are executable by the processor to generate integrated Web Service architectures for implementing integrated Web Service business systems, wherein, to generate an integrated Web Service architecture for implementing a specific integrated Web Service business system, the program instructions are executable by the processor to: identify one or more components of the integrated Web Service architecture according to one or more use case requirements for the specific integrated Web Service business system, wherein each use case requirement specifies a particular business scenario for the integrated Web Service business system; define a plurality of integration tiers and one or more Web Services technologies according to a Web Services architecture integration framework; define how each of the plurality of integration tiers communicates with others of the plurality of integration tiers according to the Web Services architecture integration framework; organize the components according to the plurality of integration tiers and two or more layers of the integrated Web Service architecture; apply one or more design patterns to the integrated Web Service architecture, wherein the one or more design patterns include one or more Web Services integration design patterns for integrating Web Services with business systems; and provide output indicating the generated integrated Web Service architecture for implementing the specific integrated Web Service business system.

Furthermore, as noted above in regard to claim 1, in paragraph [0055] Curry teaches “The framework includes a pre-built architecture that allows developers to rapidly create applications based on business components and web services.” **Curry does not teach a system comprising program instructions executable to implement a Web Services architecture design service configured to generate an integrated Web Service architecture for implementing a specific integrated Web Service business system.** Instead, Curry clearly teaches a “standard framework architecture” that clearly pre-exists and that may allegedly be used to “rapidly create applications based on business components and web services.”

On page 37 of the Response to Arguments section of the Office Action of June 24, 2010, the Office argues that “a specific instance of the framework template must be generated for the web service under development.” However, this “specific instance” would still be just a copy of Curry’s “standard framework architecture” that may be modified, and does not teach a system comprising program instructions executable to implement a Web Services architecture design service configured to generate an integrated Web Service architecture for integrating a specific Web Service with a specific business system, as recited in Applicant’s claims.

**The cited references, alone or in combination, do not teach a system as recited in Applicant’s claim 18 when the claim is viewed as a whole.**

The Office is clearly attempting a piecemeal reconstruction of Applicant’s invention in hindsight without considering the claimed invention as a whole. Such reconstruction is improper. *See, e.g., Interconnect Planning Corp. v. Feil*, 774 F.2d 1132, 1143, 227 USPQ 543, 551 (Fed. Cir. 1985) (it is insufficient to select from the prior art the separate components of the inventor’s combination, using the blueprint supplied by the inventor); *Uniroyal, Inc. v. Rudkin-Wiley Corp.*, 837 F.2d 1044, 1051-52, 5 USPQ 2d 1434, 1438 (Fed. Cir. 1988) (it is impermissible to reconstruct the claimed invention from selected pieces of prior art absent some suggestion, teaching, or motivation in the prior

art to do so). The Office cannot use the claimed invention as an instruction manual or “template” to piece together the teachings of the prior art so that the claimed invention is rendered obvious. *In re Fritch*, 23 USPQ 2d 1780, 1784 (Fed. Cir. 1992). “One cannot use hindsight reconstruction to pick and choose among isolated disclosures in the prior art to deprecate the claimed invention.” *In re Fine*, 837 F.2d 1071, 1075, 5 USPQ 2d 1596, 1600 (Fed. Cir. 1988).

**In further regard to claim 18, the Office has failed to establish that the cited references teach, program instructions executable by a processor to: identify one or more logical components of the integrated Web Service architecture according to one or more use case requirements for the specific integrated Web Service business system, wherein each use case requirement specifies a particular business scenario for the integrated Web Service business system, as recited in claim 1.**

The Office Action relies on Curry to teach these features, citing FIG. 2, paragraphs [0051]-[0052], and paragraphs [0074]-[0080]. FIG. 2 expands on element 10 (“Business Logic Development”) of Curry’s FIG. 1 (see paragraph [0074]). However, Curry does not specifically teach that element 10 (“Business Logic Development”) of Curry’s FIG. 1 is computer-implemented. Curry thus does not teach program instructions executable by the processor to identify one or more logical components of the integrated Web Service architecture according to one or more use case requirements for the specific integrated Web Service business system, wherein each use case requirement specifies a particular business scenario for the integrated Web Service business system, as recited in Applicant’s claim 18. The other cited references fail to overcome these shortcomings of Curry.

**In further regard to claim 18, the cited references do not teach at least the features of, program instructions executable by a processor to: translate the one or more use case requirements for the specific integrated Web Service business system and one or more technical constraints for the specific integrated Web Service business system to determine a plurality of Web Service components for the**

**integrated Web Service architecture, wherein the Web Service components include software components, as recited in claim 18.**

In the Office Action of June 24, 2010, the Office admits that Curry and Epionet do not teach program instructions executable by a processor to translate one or more use case requirements for a specific integrated Web Service business system and one or more technical constraints for the specific integrated Web Service business system to determine a plurality of Web Service components for an integrated Web Service architecture, wherein the Web Service components include software components. The Office asserts that Siegel teaches these features, citing page 5, last paragraph, page 6, paragraphs 1-3, and page 8, last paragraph. The Office asserts that Siegel teaches MDA tools that “generate interface definitions, application code, makefiles, and configuration files for the PSM’s middleware platform.” However, Applicant can find nothing in Siegel that specifically teaches that the MDA tools translate one or more use case requirements for a specific integrated Web Service business system and one or more technical constraints for the specific integrated Web Service business system to determine a plurality of Web Service components for an integrated Web Service architecture.

In (83), page 40, of the Response to Arguments section of the Office Action of January 20, 2011, in response to the above, the Office asserts “Siegel teaches translating use case requirements and technical constraints (i.e., UML models that specifies every detail of business functionality and behavior where it is well known in the art that use cases are one type of UML model and business functionality and behavior are considered as technical constraints) of a Web service to determine a plurality of components for the Web Service,” again citing page 5, last paragraph, page 6, paragraphs 1-3.

Applicant first notes that, even if “use cases are one type of UML model” as asserted by the Office, **Siegel does not teach that the UML model(s) described in the citation are use case models.** Furthermore, this citation states “OMG’s MDA unifies and simplifies modeling, design, implementation, and integration of applications...by

defining software fundamentally at the model level, expressed in OMG's standard Unified Modeling Language (UML). An MDA application's base model specifies every detail of its business functionality and behaviour..." Siegel is teaching that the "base model," expressed in OMG's UML specifies business functionality and behavior. Thus, even if Siegel's UML models are interpreted as "use cases," as asserted by the Office, Siegel does not teach translating one or more use case requirements for the specific integrated Web Service business system and one or more technical constraints for the specific integrated Web Service business system. Siegel's "business functionality and behavior," relied upon by the Office to teach "technical constraints," are not separately defined in Siegel, but instead are specified in Siegel's UML models, which the Office equates with Applicant's use cases. In contrast to this interpretation of the Siegel reference, claim 18 recites translating one or more use case requirements for the specific integrated Web Service business system and one or more technical constraints for the specific integrated Web Service business system, where the one or more use cases and the technical constraints are separate.

Furthermore, the Office has not provided a proper *prima facie* reason to combine the references. The Office asserts that it would have been obvious to have modified Curry/Epionet to "translate one or more use case requirements for a specific integrated Web Service business system and one or more technical constraints for the specific integrated Web Service business system to determine a plurality of Web Service components for an integrated Web Service architecture" as allegedly taught by Siegel "such that tools can automate and thereby simplify most of the building of distributed applications." However, as noted above, Siegel does not teach these features as recited in claim 18. Furthermore, the methods that are taught by Siegel (MDA tools that "generate interface definitions, application code, makefiles, and configuration files for the PSM's middleware platform.") are clearly and distinctly different than the methods taught by Curry and Epionet. The proposed modification to Curry/Epionet could not be made without changing the principle of operation of Curry's "Web services development method" or of Epionet's methods as disclosed, as Siegel teaches a completely different method than either reference. "If the proposed modification or combination of the prior

art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious.” *In re Ratti*, 270 F.2d 810, 123 USPQ 349 (CCPA 1959). Thus, one of ordinary skill would not have combined the teachings of Siegel with the teachings of Curry/Epionet in the manner proposed by the Office. Accordingly, the Office has failed to establish a *prima facie* case of obviousness.

In (83), page 46, of the Response to Arguments section of the Office Action of January 20, 2011, in response to the above, the Office asserts “the proposed modification could be made without changing the principle of operation of Curry/Epionet as Curry teaches generating use cases and the modification with Siegel would be to translate those use cases generated by Curry using the technology of Siegel to simplify most of the building of distributed applications.” Applicant notes that the Office’s assertion only serves to make it clear that the proposed modification would change the principle of operation of Curry/Epionet. Siegel teaches a completely different method than either reference. “Using the technology of Siegel to simplify most of the building of distributed applications” as proposed by the Office would clearly change the principle of operation of Curry/Epionet.

**In further regard to claim 18, the cited references do not teach at least the features of, program instructions executable by a processor to: categorize the Web Service components into two or more related groups according to a Web Services architecture integration framework.**

The Office cites Curry, paragraph [0164], “categorization of components so that a search engine can be used for efficient retrieval,” as allegedly teaching categorizing the Web Service components into two or more related groups according to a Web Services architecture integration framework. However, this paragraph appears in a section titled “Rework” that begins at paragraph [0159]. Paragraph [0160] states “Once the application is completed, a Rework step 34, (see FIG. 1) can be performed to place the newly developed web services in form for addition to the Web Services Master Library.”

Paragraph [0161] states “adjudication is performed to determine which of the assets, i.e., classes, components, prototypes, applications etc., are in some way suitable for reuse.” Paragraph [0164] states “After testing, a categorization step 206 is performed to sort or index the components so that later discovery and retrieval will be optimized” and “Once the components are categorized they can be published to the Master Library.” Clearly, the “categorization step” for sorting or indexing components to be put into a master library as taught in paragraph [0164] has nothing to do with categorizing Web Service components into two or more related groups according to a Web Services architecture integration framework as part of generating an integrated Web Service architecture for implementing a specific integrated Web Service business system, as recited in claim 18.

In (84), pages 46-47, of the Response to Arguments section of the Office Action of January 20, 2011, in response to the above, the Office asserts “it is not clear to the Examiner how the categorization step of Curry has nothing to do with the categorization of the instant claim. Examiner speculates that the Applicant is arguing that the categorization step of Curry is directed to reusing the Web Service component while the limitation does not intend reuse as the purpose of categorization.” Applicant is not arguing that the categorization step of Curry is directed to “reusing the Web Service components.” Applicant is pointing out that, according to the plain language of Curry, Curry’s categorization step is performed to “sort or index the components so that later discovery and retrieval will be optimized” (paragraph [0164]). In contrast, claim 18 recites categorizing the Web Service components into two or more related groups according to a Web Services architecture integration framework. Claim 18 later recites organizing the groups of Web Service components according to the plurality of integration tiers and two or more layers of the integrated Web Service architecture. According to the plain language of Applicant’s claim, these limitations are performed in generating an integrated Web Service architecture for implementing a specific integrated Web Service business system.

The Office goes on to assert “the categorization step of Curry still teaches the categorization limitation as recited since the categorize web service components is

performed as part of the development process of the Web Service regardless of whether the purpose of categorization in Curry is to index the component in a library for reuse.” Applicant traverses this assertion for at least the reason that paragraph [0160] states “Once the application is completed, a Rework step 34...can be performed.” Curry’s “categorization step 206” is part of this “rework step 34”, which Curry explicitly states is performed “once the application is completed.” Thus, contrary to the Office’s assertion Curry’s categorization step clearly is not performed as part of a “development process of a Web Service.”

**In further regard to claim 18, the Office has failed to establish that the cited references teach, program instructions executable by a processor to: define a plurality of integration tiers for the integrated Web Service architecture and one or more Web Services technologies for the integrated Web Service architecture according to the Web Services architecture integration framework; define how each of the plurality of integration tiers communicates with others of the plurality of integration tiers in the integrated Web Service architecture according to the Web Services architecture integration framework; and organize the groups of Web Service components according to the plurality of integration tiers and two or more layers of the integrated Web Service architecture.**

The Office Action relies on Curry to teach these features, citing paragraphs [0056-59], [0136], and [0151]. However, Curry only teaches that the “framework structure” may include layers, and does not teach organizing the groups of Web Service components according to the plurality of integration tiers and two or more layers of the integrated Web Service architecture. Curry only teaches that the framework may include a number of layers, for example in paragraph [0056]: “The structure includes a number of layers or sub-architectures. Each layer addresses a specific area within a standard web application.” Paragraph [0151] refers to “the logical layers of the application,” which appear to be the same “layers” referred to in paragraph [0056]. Curry does not teach organizing a groups of Web Service components according to a plurality of integration



tiers and two or more layers of the integrated Web Service architecture, as recited in claim 18.

In addition, the Office relied on Curry, paragraph [0164], “categorization of components so that a search engine can be used for efficient retrieval,” as allegedly teaching categorizing the Web Service components into two or more related groups according to a Web Services architecture integration framework. Curry clearly does not teach that these categorized components as taught at paragraph [0164] are organized according to a plurality of integration tiers and two or more layers of the integrated Web Service architecture, as recited in claim 18.

**In further regard to claim 18, the Office has failed to establish a proper *prima facie* reason to combine the references.**

In regard to the proposed combination of Curry and Huang, the Office Action asserts that “it would have been obvious...to have modified Curry such that the web service architecture is generated in accordance with one or more integration design patterns as taught by Huang because design patterns are well known in the art and commonly used by programmers since design patterns provide the benefit of capturing a standard solution to a common programming problem for reuse.” However, Huang teaches four specific design patterns – a Composite pattern, a Mediator pattern, a Command pattern, and a State pattern. It is unclear as to how Curry’s system would be modified with these specific design patterns, much less how the modification could be made without changing the principle of operation of Curry’s “Web services development method” as disclosed. Curry and Huang each propose distinctly different methods for achieving similar results. Modifying Curry with Huang’s specific “design patterns” would appear to necessitate significant changes in Curry’s system. “If the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious.” *In re Ratti*, 270 F.2d 810, 123 USPQ 349 (CCPA 1959). Thus, one of ordinary skill would not have combined the teachings of

Huang with the teachings of Curry in the manner proposed by the Office. Accordingly, the Office has failed to establish a *prima facie* case of obviousness.

In the Response to Arguments, 77, on pages 37-38 of the Office Action of June 24, 2010, the Office asserts that it is “not clear...why incorporating the use of design patterns in the development method taught by Curry would necessitate significant changes.” However, as Applicant has noted, Huang teaches four specific design patterns – a Composite pattern, a Mediator pattern, a Command pattern, and a State pattern. These design patterns are specifically intended for use in Huang’s “Web services-based framework for business integration solutions.” (Title). Curry and Huang each propose distinctly different methods for achieving similar results. It is unclear as to how Curry’s system would be modified with these specific design patterns, much less how the modification could be made without changing the principle of operation of Curry’s “Web services development method” as disclosed. Modifying Curry with Huang’s specific “design patterns” would thus appear to necessitate significant changes in Curry’s system. Moreover, while the Office broadly asserts that it is obvious that “one of ordinary skill in the art...would have the necessary skills to use design patterns for that particular problem,” the Office provides no support as to how Curry’s system would be modified with these specific design patterns, much less how the modification could be made without changing the principle of operation of Curry’s “Web services development method” as disclosed.

In regard to the Curry and Epiowave references, the Office Action asserts it would have been obvious that “the system of Curry would include a Web Services architecture design service since the Epiowave toolset used in Curry is capable of planning, prototyping, testing, developing, and deploying web services.” However, at paragraph [0072] Curry states that “In an illustrative embodiment of the invention, the Epiowave™ environment available from the applicant, Epionet Corporation of Dublin, Ireland, serves as the D&D environment in the D&D application creation step 32.” Curry does not describe that other aspects of Curry’s system relied upon by the Office are performed using Epiowave. Moreover, the Office has not explained how a combination of

Epiowave with Curry would result in anything but the system that Curry already teaches. Accordingly, the Office has failed to establish a *prima facie* case of obviousness.

In (85), pages 47, of the Response to Arguments section of the Office Action of January 20, 2011, the Office asserts “As per the rest of the arguments for independent claims 18, 31, and 42, the arguments presented are similar to those presented for claim 1. Therefore, the arguments are not persuasive for the same reasons as those presented for claim 1.” Applicant refers to Applicant’s traversals of the Office’s reasons as presented regarding claim 1 in the Response to Arguments, which apply equally to claim 18.

Thus, for at least the reasons presented above, the rejection of claim 18 is not supported by the cited art and removal thereof is respectfully requested. Similar remarks as those above regarding claim 18 also apply to claims 58 and 82.

Applicant also asserts that the rejection of numerous ones of the dependent claims under the 35 U.S.C. § 103(a) rejection is further unsupported by the cited art. However, since the rejection has been shown to be unsupported for the independent claims, a further discussion of the dependent claims is not necessary at this time.

The Office Action on page 25 rejected claims 25, 30, 65 and 89 under 35 U.S.C. § 103(a) as allegedly being unpatentable over Curry in view of Epionet and further in view of Huang, Olsen and Chappell. However, since the rejection has been shown to be unsupported for the independent claims from which these claims depend, a further discussion of this rejection is not necessary at this time.

The Office Action on pages 26-27 rejected claims 26, 66 and 90 under 35 U.S.C. § 103(a) as allegedly being unpatentable over Curry in view of Epionet and further in view of Siegel, Huang, Olsen and Connell. However, since the rejection has been shown to be unsupported for the independent claims from which these claims depend, a further discussion of this rejection is not necessary at this time.

The Office Action on pages 27-28 rejected claims 27, 67 and 91 under 35 U.S.C. § 103(a) as allegedly being unpatentable over Curry in view of Epionet and further in view of Siegel, Huang, Olsen and Curtis.

The Office Action on pages 28-29 rejected claims 31, 34, 35, 37-39 and 41 under 35 U.S.C. § 103(a) as allegedly being unpatentable over Curry in view of Curtis and further in view of Epionet and Huang. Applicant traverses this rejection for at least the reasons indicated below.

**In regard to claim 31, as a first matter, the cited references, alone or in combination, do not teach an integrated Web Services business system configured and implemented according to an integrated Web Service architecture generated by a computer-implemented integrated Web Services architecture design service according to a structured integration methodology for designing and generating integrated Web Service architectures for integrating Web Services technologies with business systems comprising heterogeneous components, as recited in claim 31.**

As noted above in regards to claim 1, Curry does not teach that the method illustrated in FIG. 1 is a computer-implemented method, and Curry does not describe a computer system that implements the method illustrated in FIG. 1. At most, Curry describes that particular steps or sub-steps of Curry's method may be assisted by or performed using existing software tools. Huang only conceptually describes a framework for developing Web services-based business integration solutions, and does not teach or even suggest a computer implementation of the framework for developing Web services-based business integration solutions. The Epionet/Epionetwave references may broadly assert that the toolset may provide for "planning, prototyping, testing, developing, and deploying web services," however, the references do not teach that the Epionet/Epionetwave platform performs the elements of the subject matter as recited in claim 31. Curtis, on the other hand, is directed to "customer relationship management architectures" (see, e.g., Curtis, Title, Abstract, paragraph [0001] ("The present invention relates to the field of customer relationship management. More particularly, the present invention, in various

specific embodiments, involves methods and systems directed to providing a customer relationship management architecture.”) Curtis never mentions web services, and is not directed to web services, and does not teach or suggest is not directed to designing and generating integrated Web Service architectures for integrating Web Services technologies with business systems comprising heterogeneous components.

On page 8 of the Office Action of June 24, 2010, the Office admits that Curry does not teach implementing a Web Services architecture design service to generate integrated Web Service architectures for integrating Web Services with business systems, and asserts that it would have been obvious that “the system of Curry would include a Web Services architecture design service since the Epiowave toolset used in Curry is capable of planning, prototyping, testing, developing, and deploying web services.” The Office further argues this assertion in the Response to Arguments, 72, on pages 36-37. **However, as noted above, Curry only describes that the “the D&D application creation step 32” of FIG. 1 may be performed using Epiowave.** Curry does not describe that other aspects of Curry’s system relied upon by the Office are performed using Epiowave or any other automated system. Moreover, the Office has not explained how a combination of Epiowave with Curry would result in anything but the system that Curry already teaches, for example as illustrated in FIG. 1. Neither Curry nor any combination of the cited art teaches a computer-implemented system that implements all of the elements of the method as illustrated in Curry’s FIG. 1, nor does Curry or any combination of the cited art teach a Web Services architecture design service that performs all of the elements as recited in claim 18 of the instant application.

In contrast to the cited references, claim 31 recites an integrated Web Services business system is configured and implemented according to an integrated Web Service architecture generated by a computer-implemented integrated Web Services architecture design service according to a structured integration methodology for designing and generating integrated Web Service architectures for integrating Web Services technologies with business systems comprising heterogeneous components such that: the plurality of heterogeneous business components are organized according to a plurality of

integration tiers and two or more layers of the integrated Web Service architecture; and one or more design patterns including one or more Web Services integration design patterns for integrating Web Services with business systems are applied to the integrated Web Service architecture, wherein each design pattern models a particular structure that is applicable to the integrated Web Service.

Furthermore, as noted above in regard to claim 1, in paragraph [0055] Curry teaches “The framework includes a pre-built architecture that allows developers to rapidly create applications based on business components and web services.” **Curry does not teach a computer-implemented integrated Web Services architecture design service for generating integrated Web Service architectures as recited in claim 31.** Instead, Curry clearly teaches a “standard framework architecture” that clearly pre-exists and that may allegedly be used to “rapidly create applications based on business components and web services.”

On page 37 of the Response to Arguments section of the Office Action of June 24, 2010, the Office argues that “a specific instance of the framework template must be generated for the web service under development.” However, this “specific instance” would still be just a copy of Curry’s “standard framework architecture” that may be modified, and does not teach a computer-implemented integrated Web Services architecture design service for generating integrated Web Service architectures as recited in claim 31.

**The cited references, alone or in combination, do not teach a system as recited in Applicant’s claim 31 when the claim is viewed as a whole.**

The Office is clearly attempting a piecemeal reconstruction of Applicant’s invention in hindsight without considering the claimed invention as a whole. Such reconstruction is improper. *See, e.g., Interconnect Planning Corp. v. Feil*, 774 F.2d 1132, 1143, 227 USPQ 543, 551 (Fed. Cir. 1985) (it is insufficient to select from the prior art the separate components of the inventor’s combination, using the blueprint supplied by

the inventor); *Uniroyal, Inc. v. Rudkin-Wiley Corp.*, 837 F.2d 1044, 1051-52, 5 USPQ 2d 1434, 1438 (Fed. Cir. 1988) (it is impermissible to reconstruct the claimed invention from selected pieces of prior art absent some suggestion, teaching, or motivation in the prior art to do so). The Office cannot use the claimed invention as an instruction manual or “template” to piece together the teachings of the prior art so that the claimed invention is rendered obvious. *In re Fritch*, 23 USPQ 2d 1780, 1784 (Fed. Cir. 1992). “One cannot use hindsight reconstruction to pick and choose among isolated disclosures in the prior art to deprecate the claimed invention.” *In re Fine*, 837 F.2d 1071, 1075, 5 USPQ 2d 1596, 1600 (Fed. Cir. 1988).

**In further regard to claim 31, the cited art does not teach at least the features of wherein the integrated Web Services business system is configured and implemented according to an integrated Web Service architecture generated by a computer-implemented integrated Web Services architecture design service according to a structured integration methodology for designing and generating integrated Web Service architectures for integrating Web Services technologies with business systems comprising heterogeneous components such that: the plurality of heterogeneous business components are organized according to the plurality of integration tiers and two or more layers of the integrated Web Service architecture, as recited in claim 31.**

The Office Action relies on Curry to support the assertion that the cited art teaches a plurality of tiers of an integrated Web Services business system, wherein the plurality of tiers comprises a presentation tier, a business tier, and a resources tier, citing “automating layer separability”, paragraphs [0016] and [0028]. The Office admits that Curry does not teach wherein the plurality of integration tiers also comprise a client tier and an integration tier, and relies on Curtis to teach these features, asserting “Curtis is cited to teach a method for separating an integrated management architecture into tiers, including a client tier, a presentation tier, a business tier, an integration tier, and a resources tier,” citing paragraphs [0007] and [0028]-[0033]. The Office then contends “It would have been obvious...to have modified Curry such that the plurality of tiers also

include a client tier and an integration tier because the n-tier architecture for web services is well known in the art (see [0016] of Curry) and separating web services into a tier structure that includes a client tier and an integration tier is also well known (see [0007] of Curtis).”

However, claim 31 recites that the plurality of heterogeneous business components of the integrated Web Services business system are organized according to the plurality of integration tiers and two or more layers of the integrated Web Service architecture. Applicants contend that an integrated Web Service architecture including a plurality of integration tiers and two or more layers was not known in the art.

Furthermore, Curtis is directed to “customer relationship management architectures” (see, e.g., Curtis, Title, Abstract, paragraph [0001] (“The present invention relates to the field of customer relationship management. More particularly, the present invention, in various specific embodiments, involves methods and systems directed to providing a customer relationship management architecture.”) Curtis never mentions web services. Thus, the Office’s contention that “separating web services into a tier structure that includes a client tier and an integration tier is also well known” is not supported by Curtis.

**In further regard to claim 31, the Office has failed to establish a proper *prima facie* reason to combine the references.**

In regard to the proposed combination of Curry and Huang, the Office Action asserts that “it would have been obvious...to have modified Curry such that the web service architecture is generated in accordance with one or more integration design patterns as taught by Huang because design patterns are well known in the art and commonly used by programmers since design patterns provide the benefit of capturing a standard solution to a common programming problem for reuse.” However, Huang teaches four specific design patterns – a Composite pattern, a Mediator pattern, a Command pattern, and a State pattern. It is unclear as to how Curry’s system would be



modified with these specific design patterns, much less how the modification could be made without changing the principle of operation of Curry's "Web services development method" as disclosed. Curry and Huang each propose distinctly different methods for achieving similar results. Modifying Curry with Huang's specific "design patterns" would appear to necessitate significant changes in Curry's system. "If the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious." *In re Ratti*, 270 F.2d 810, 123 USPQ 349 (CCPA 1959). Thus, one of ordinary skill would not have combined the teachings of Huang with the teachings of Curry in the manner proposed by the Office. Accordingly, the Office has failed to establish a *prima facie* case of obviousness.

In the Response to Arguments, 77, on pages 37-38 of the Office Action of June 24, 2010, the Office asserts that it is "not clear...why incorporating the use of design patterns in the development method taught by Curry would necessitate significant changes." However, as Applicant has noted, Huang teaches four specific design patterns – a Composite pattern, a Mediator pattern, a Command pattern, and a State pattern. These design patterns are specifically intended for use in Huang's "Web services-based framework for business integration solutions." (Title). Curry and Huang each propose distinctly different methods for achieving similar results. It is unclear as to how Curry's system would be modified with these specific design patterns, much less how the modification could be made without changing the principle of operation of Curry's "Web services development method" as disclosed. Modifying Curry with Huang's specific "design patterns" would thus appear to necessitate significant changes in Curry's system. Moreover, while the Office broadly asserts that it is obvious that "one of ordinary skill in the art...would have the necessary skills to use design patterns for that particular problem," the Office provides no support as to how Curry's system would be modified with these specific design patterns, much less how the modification could be made without changing the principle of operation of Curry's "Web services development method" as disclosed.

In regard to the Curry and Epiowave references, the Office Action asserts it would have been obvious that “the system of Curry would include a Web Services architecture design service since the Epiowave toolset used in Curry is capable of planning, prototyping, testing, developing, and deploying web services.” However, at paragraph [0072] Curry states that “In an illustrative embodiment of the invention, the Epiowave™ environment available from the applicant, Epionet Corporation of Dublin, Ireland, serves as the D&D environment in the D&D application creation step 32.” Curry does not describe that other aspects of Curry’s system relied upon by the Office are performed using Epiowave. Moreover, the Office has not explained how a combination of Epiowave with Curry would result in anything but the system that Curry already teaches. Accordingly, the Office has failed to establish a *prima facie* case of obviousness.

In (85), pages 47, of the Response to Arguments section of the Office Action of January 20, 2011, the Office asserts “As per the rest of the arguments for independent claims 18, 31, and 42, the arguments presented are similar to those presented for claim 1. Therefore, the arguments are not persuasive for the same reasons as those presented for claim 1.” Applicant refers to Applicant’s traversals of the Office’s reasons as presented regarding claim 1 in the Response to Arguments, which apply equally to claim 31.

Thus, for at least the reasons presented above, the rejection of claim 31 is not supported by the cited art and removal thereof is respectfully requested.

Applicant also asserts that the rejection of numerous ones of the dependent claims under the 35 U.S.C. § 103(a) rejection is further unsupported by the cited art. However, since the rejection has been shown to be unsupported for the independent claims, a further discussion of the dependent claims is not necessary at this time.

The Office Action on page 34 rejected claim 32 under 35 U.S.C. § 103(a) as allegedly being unpatentable over Curry in view of Curtis and further in view of Epionet, Huang and Connell. However, since the rejection has been shown to be unsupported for

the independent claim from which this claim depends, a further discussion of this rejection is not necessary at this time.

The Office Action on pages 34-35 rejected claims 36 and 40 under 35 U.S.C. § 103(a) as allegedly being unpatentable over Curry in view of Curtis, Epionet, Huang and Chappell. However, since the rejection has been shown to be unsupported for the independent claims from which these claims depend, a further discussion of this rejection is not necessary at this time.

The Office Action on page 36 rejected claims 42 and 43 under 35 U.S.C. § 103(a) as allegedly being unpatentable over Curry in view of Huang and further in view of Olsen. Applicant traverses this rejection for at least the reasons indicated below.

**In regard to claim 42, as a first matter, neither of the cited references (Curry and Huang) teach computer-implemented means for generating integrated Web Service architectures.**

As noted above in regards to claim 1, Curry does not teach that the method illustrated in FIG. 1 is a computer-implemented method, and Curry does not describe a computer system that implements the method illustrated in FIG. 1. At most, Curry describes that particular steps or sub-steps of Curry's method may be assisted by or performed using existing software tools. Huang only conceptually describes a framework for developing Web services-based business integration solutions, and does not teach or even suggest a computer implementation of the framework for developing Web services-based business integration solutions. Olsen only teaches that generic web services are customized according to input information and does not teach program instructions executable by a processor to generate integrated Web Service architectures for integrating Web Services with business systems.

In contrast to the cited art, claim 42 recites computer-implemented means for generating an integrated Web Services architecture for integrating a specific Web Service with a specific business system.

Furthermore, as noted above in regard to claim 1, in paragraph [0055] Curry teaches “The framework includes a pre-built architecture that allows developers to rapidly create applications based on business components and web services.” **Curry does not teach an integrated Web Service architecture that is generated by computer-implemented means for generating an integrated Web Services architecture for integrating a Web Service with a business system comprising a plurality of heterogeneous components.** Instead, Curry only teaches a “standard framework architecture” that clearly pre-exists and that may allegedly be used to “rapidly create applications based on business components and web services.”

On page 37 of the Response to Arguments section of the Office Action of June 24, 2010, the Office argues that “a specific instance of the framework template must be generated for the web service under development.” However, this “specific instance” would still be just a copy of Curry’s “standard framework architecture” that may be modified, and does not teach a system comprising program instructions executable to implement a Web Services architecture design service configured to generate an integrated Web Service architecture for integrating a specific Web Service with a specific business system, as recited in Applicant’s claims.

**The cited references, alone or in combination, do not teach a system for integrating Web Services with business systems as recited in Applicant’s claim 42 when the claim is viewed as a whole.**

The Office is clearly attempting a piecemeal reconstruction of Applicant’s invention in hindsight without considering the claimed invention as a whole. Such reconstruction is improper. *See, e.g., Interconnect Planning Corp. v. Feil*, 774 F.2d 1132, 1143, 227 USPQ 543, 551 (Fed. Cir. 1985) (it is insufficient to select from the prior art

the separate components of the inventor's combination, using the blueprint supplied by the inventor); *Uniroyal, Inc. v. Rudkin-Wiley Corp.*, 837 F.2d 1044, 1051-52, 5 USPQ 2d 1434, 1438 (Fed. Cir. 1988) (it is impermissible to reconstruct the claimed invention from selected pieces of prior art absent some suggestion, teaching, or motivation in the prior art to do so). The Office cannot use the claimed invention as an instruction manual or "template" to piece together the teachings of the prior art so that the claimed invention is rendered obvious. *In re Fritch*, 23 USPQ 2d 1780, 1784 (Fed. Cir. 1992). "One cannot use hindsight reconstruction to pick and choose among isolated disclosures in the prior art to deprecate the claimed invention." *In re Fine*, 837 F.2d 1071, 1075, 5 USPQ 2d 1596, 1600 (Fed. Cir. 1988).

**In further regard to claim 42, the cited art fails to teach at least the features of computer-implemented means for applying a Web Services structured methodology and one or more design patterns to the generated integrated Web Service architecture to identify a plurality of integrated Web Service components including one or more of the business system components and to organize the integrated Web Service components according to the integrated Web Service architecture, wherein the plurality of integrated Web Service components are organized according to two or more integration tiers and two or more layers of the integrated Web Service architecture, as recited in claim 42.**

Curry only teaches a framework that may include a number of layers, for example in paragraph [0056]: "The structure includes a number of layers or sub-architectures. Each layer addresses a specific area within a standard web application." Curry does not teach a plurality of integrated Web Service components organized according to two or more integration tiers and two or more layers of the integrated Web Service architecture, as recited in claim 42. The other cited references fail to overcome this shortcoming.

**In further regard to claim 42, the Office has failed to establish a proper *prima facie* reason to combine the Curry and Huang references.**

The Office Action asserts that “it would have been obvious...to have modified Curry such that the web service architecture is generated in accordance with one or more integration design patterns as taught by Huang because design patterns are well known in the art and commonly used by programmers since design patterns provide the benefit of capturing a standard solution to a common programming problem for reuse.” However, Huang teaches four specific design patterns – a Composite pattern, a Mediator pattern, a Command pattern, and a State pattern. It is unclear as to how Curry’s system would be modified with these specific design patterns, much less how the modification could be made without changing the principle of operation of Curry’s “Web services development method” as disclosed. Curry and Huang each propose distinctly different methods for achieving similar results. Modifying Curry with Huang’s specific “design patterns” would appear to necessitate significant changes in Curry’s system. “If the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious.” *In re Ratti*, 270 F.2d 810, 123 USPQ 349 (CCPA 1959). Thus, one of ordinary skill would not have combined the teachings of Huang with the teachings of Curry in the manner proposed by the Office. Accordingly, the Office has failed to establish a *prima facie* case of obviousness.

In the Response to Arguments, 77, on pages 37-38 of the Office Action of June 24, 2010, the Office asserts that it is “not clear...why incorporating the use of design patterns in the development method taught by Curry would necessitate significant changes.” However, as Applicant has noted, Huang teaches four specific design patterns – a Composite pattern, a Mediator pattern, a Command pattern, and a State pattern. These design patterns are specifically intended for use in Huang’s “Web services-based framework for business integration solutions.” (Title). Curry and Huang each propose distinctly different methods for achieving similar results. It is unclear as to how Curry’s system would be modified with these specific design patterns, much less how the modification could be made without changing the principle of operation of Curry’s “Web services development method” as disclosed. Modifying Curry with Huang’s specific “design patterns” would thus appear to necessitate significant changes in Curry’s system.

Moreover, while the Office broadly asserts that it is obvious that “one of ordinary skill in the art...would have the necessary skills to use design patterns for that particular problem,” the Office provides no support as to how Curry’s system would be modified with these specific design patterns, much less how the modification could be made without changing the principle of operation of Curry’s “Web services development method” as disclosed.

In (85), pages 47, of the Response to Arguments section of the Office Action of January 20, 2011, the Office asserts “As per the rest of the arguments for independent claims 18, 31, and 42, the arguments presented are similar to those presented for claim 1. Therefore, the arguments are not persuasive for the same reasons as those presented for claim 1.” Applicant refers to Applicant’s traversals of the Office’s reasons as presented regarding claim 1 in the Response to Arguments, which apply equally to claim 42.

Thus, for at least the reasons presented above, the rejection of claim 42 is not supported by the cited art and removal thereof is respectfully requested.

Applicant also asserts that the rejection of numerous ones of the dependent claims under the 35 U.S.C. § 103(a) rejection is further unsupported by the cited art. However, since the rejection has been shown to be unsupported for the independent claims, a further discussion of the dependent claims is not necessary at this time.

The Office Action on page 39 rejected claim 44 under 35 U.S.C. § 103(a) as allegedly being unpatentable over Curry in view of Huang and further in view of Olsen and Chappell. However, since the rejection has been shown to be unsupported for the independent claims from which these claims depend, a further discussion of this rejection is not necessary at this time.

## CONCLUSION

Applicants submit the application is in condition for allowance, and an early notice to that effect is respectfully requested.

If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5681-66303/RCK.

Respectfully submitted,

/Robert C. Kowert/

Robert C. Kowert, Reg. #39,255  
Attorney for Applicants

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.  
P.O. Box 398  
Austin, TX 78767-0398  
Phone: (512) 853-8850

Date: April 20, 2011